ÖREBRO UNIVERSITET
INSTITUTIONEN FÖR
NATURVETENSKAP OCH TEKNIK

Kursens namn	**Inbyggda system för civilingenjörer – DT511G**
Examinationsmomentets namn/provkod	**0100**
Datum	2019-05-29
Tid	Kl. 08:15 – 11:15

Tillåtna hjälpmedel	Calculator
Instruktion	Read all the questions carefully. Start answering each question in a new answer sheet. Write only on one side of an answer sheet. Write the exam code on every answer sheet. Write in a readable way!
Viktigt att tänka på	
Ansvarig/-a lärare (ev. telefonnummer)	Farhang Nemati Tel: 019-303264 Mobil: 0702533418
Totalt antal poäng	40
Betyg (ev. ECTS)	The exam contains 5 questions. The total points are 40. At least 20 points are required for grade 3 (pass), 30 points for grade 4 and 35 points for grade 5
Tentamensresultat	The results will be notified in Studentforum within 15 working days after the exam.
Övrigt	You can write your answers in English or Swedish.

Good Luck!

1. (10 points)

Briefly answer the following questions (You may answer a question by an example or a drawing if it makes sense):

a) What is Cross-Platform Development?
b) What are the two types of Real-Time Systems?
c) What is the difference between a periodic and an aperiodic task?
d) What is a Monolithic way of executing a set of tasks of a real-time system?
e) One of the properties of a system that can be checked using formal models like Petri net is Liveness. What does Liveness mean?
f) What does preemption of a task mean?
g) What is a Monitor?
h) What is a mutex?
i) What does it mean to say that a resource has “mutually exclusive access”?
j) When does a deadlock happen?

2. (10 points)

You are required to write a program for an alarm system to be installed at home. Your system will include 1 fire detector, 1 motion sensor, and 1 door sensor (to sense if the door is open or closed). All the 3 sensors communicate to a central computer on which your program will be running. If anything unusual happens (for example if fire is detected or the door is open), the computer sends a message to the user (home owner).

In the program you have to have 4 periodic tasks:

- *task1* receives information from the fire detector every 2 seconds. In case fire is detected by the sensor, the task will set a flag to on (for example using a global variable where value 1 means on and 0 means off).
- *task2* receives information from the motion sensor every 500 milliseconds. In case any motion is detected by the sensor the task sets a flag to on.
- *task3* receives information from the door sensor every 400 milliseconds and if the door is open a flag is set to on.
- *task4* will run with a period of 5 seconds. It sends all 3 flags (fire, motion, and door flags) in a message to the user if at least one of them is set to on.

Notice: Every global variable that can be accessed by more than one task has to be protected so that not more than one task can have access to it at the same time.

Notice: You can use the following function calls in your implementation:

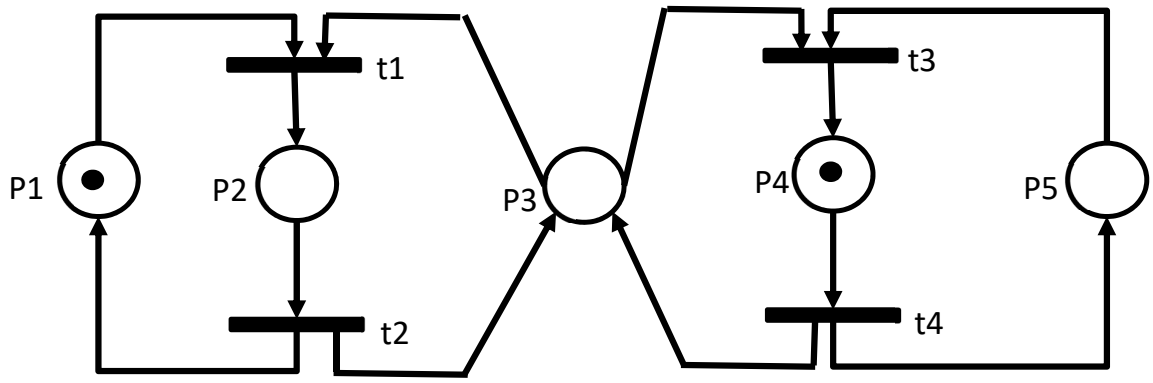
int fireSignal(); // **reads the fire detector and returns 1 if fire is detected**
int motionSignal(); // **reads the motion sensor and returns 1 if a motion is detected**
int doorSignal(); // **reads the door sensor and returns 1 if the door is open**
void sendMessage(int fire, int motion, int door); // **sends a message to the user with the values of the 3 flags set by task1, task2, and task3**
void vTaskDelay(pdMS_TO_TICKS(ms)); // **the caller task is delayed for *ms* milliseconds**

```
BaseType_t xTaskCreate( TaskFunction_t pvTaskCode, char * const pcName,
                        configSTACK_DEPTH_TYPE usStackDepth,
                        void *pvParameters,
                        UBaseType_t uxPriority,
                        TaskHandle_t *pxCreatedTask
                      );// creates a task
SemaphoreHandle_t xSemaphoreCreateMutex( void );  // creates a mutex
xSemaphoreTake( SemaphoreHandle_t xSemaphore, TickType_t xTicksToWait ); // locks a mutex
xSemaphoreGive( SemaphoreHandle_t xSemaphore ); // unlocks a mutex
```

3. (6 points)

Given the following Petri net model,

a) Draw the reachability graph of the Petri net
b) Are markings (0, 1, 0, 1, 0) and (0, 1, 0, 0, 1) reachable?



4. (8 points)

a) How does Earliest Deadline First (EDF) work for a periodic independent preemptive task set? How to check if the task set is schedulable by EDF? **(3 points)**
b) How does Priority Inheritance Protocol (PIP) work? **(2 points)**
c) What is a Directed Acyclic Graph (DAG) used for? **(1 point)**
d) What is Jitter? **(2 points)**

5. (6 points)

Assume we have the following task set where all tasks arrive at the same time. The precedence dependency among the tasks is shown by the DAG. Find a schedule using Latest Deadline First (LDF). Is the schedule a feasible schedule?

	e_i	d_i
τ_1	1	2
τ_2	1	4
τ_3	1	6
τ_4	1	6
τ_5	1	5
τ_6	1	7